# Practice-Relevant Pedagogy for Mining Software Engineering Curricula Assets

Rajiv Ramnath
*Collaborative for Enterprise Transformation and Innovation (CETI)*
*Dept. of Computer Science and Engineering*
*The Ohio State University, Columbus, Ohio*
ramnath.6@osu.edu

Jay Ramanathan
*CETI*
*Dept. of Computer Science and Engineering*
*The Ohio State University, Columbus, Ohio*
jayram@cse.ohio-state.edu

Umesh Bellur
*Department of Computer Science and Engineering*
*Indian Institute of Technology, Bombay, India*
umesh.bellur@iitb.ac.in

OSU-CISRC-5/07-TR36
6/20/2007

## 1. Overview

Software Engineering (SE) is a critical discipline in creating, assembling and operating complex information systems. However, industry software engineering practice and needs have significantly outpaced and diverged from the standard software engineering academic curriculum. Academic emphasis [SWEBOK04] continues to be on: a) the development of software from scratch rather than integration, evolution, deployment on complex shared infrastructures, b) primarily functional requirements, as compared to the emphasis on non-functional attributes required of software today, c) technical aspects alone rather than Business-IT perspectives, d) management skills taught outside of the IT context, and e) co-located development rather than global collaborative development targeted at global markets. The result is that starting from an entry-level; professionals inefficiently learn missing aspects of their trade primarily through experiential learning. Thus, industry cannot count on baseline knowledge before assigning projects. Academic bodies of knowledge like the Computing Curricula 2005 [CC2005] have attempted to define the areas of computing fundamentals with a view to providing flexibility in creating tailored programs. Despite this body of work, there is still research that needs to be conducted into what to bring into the classroom in terms of software engineering as the software application landscape continues to evolve. We posit that a learning-outcome-driven, reflective-practice-based pedagogical approach to curriculum development is critical to identifying knowledge assets and frameworks. We support our hypothesis through findings from over twenty-five projects drawn from two categories of field research from two Universities over a period of four years. This paper is organized as follows: Section 2 describes challenges in software engineering curriculum enhancement by examining trends in the practice of computing, Section 3 describes our pedagogical approach to addressing this challenge, Section 4 provides examples of where we have been successful in our approach and Section 5 summarizes, concludes and describes future work. This paper thus attempts to provide a *process* for understanding *where* to target curriculum knowledge mining for software engineering, how to *mine* and develop knowledge, how to establish *relevance,* and how to achieve results within shorter time frames than through a traditional research process. We also present critical success factors, practical implications, limitations of this approach and future work planned.

## 2. The Challenges of SE (Software Engineering) Curriculum Enhancement

The problem is best captured in "Educating the Reflective Practitioner" [Schon87], where Donald Schon writes: "In the varied topography of professional practice, there is a high ground overlooking a swamp. On the high ground, manageable (but relatively unimportant) problems lend themselves to solutions through the application of research-based theory and technique. In the swampy lowland, messy, confusing (but important) problems defy technical solution." Delivering scalable, flexible, adaptable Information Technology (IT) services is the quintessential example of working in the swampy lowland, where traditional academic approaches fail to deliver value. This is manifested in the majority of projects that are unsuccessful [STANDISH] and an increasing reliance on external consultants even for projects where internal, often proprietary, domain knowledge was the primary driver of the business rules in the system[1].

In response to evolution in the field of practice, the typical approach within academia has been to move slowly and deliberately, and wait for applicable knowledge to emerge through traditional research, if at all. Finding knowledge that is applicable takes considerable time - upwards of decades - and in the meantime, with the acceleration in application and adoption of technology industry and academia will have continued to diverge. In other words, we have to find ways to address a moving target.

To understand and address important aspects of this challenge, we first explore key *workforce trends* and root causes for curriculum obsolescence. These trends allow us to target our mining efforts. In the subsequent sections we provide a framework for addressing the issues, followed by practical examples of developing curricula and relevance.

### 2.1. Trend: Shift from Technology Producing to Technology Using Companies

Just a decade ago, the majority of IT professionals were employed in the technology producing industry – at IBM, Microsoft, DEC, ComputerVision, Unisys and so on. No longer is this the case. Beginning with the run-up to Y2K, the landscape has *shifted* to where IT employment in the technology-consuming service industry (comprising all those organizations that use information technology to enable their actual business - such as banks, airlines, and retail) dominates. *Services,* in general, now account for 80% of the U.S. gross domestic product, and service workers also form the primary users of IT. Finally realizing this shift, strategic initiatives underway, such as those within the European Union[2], have already made service-related IT issues an imperative.

---

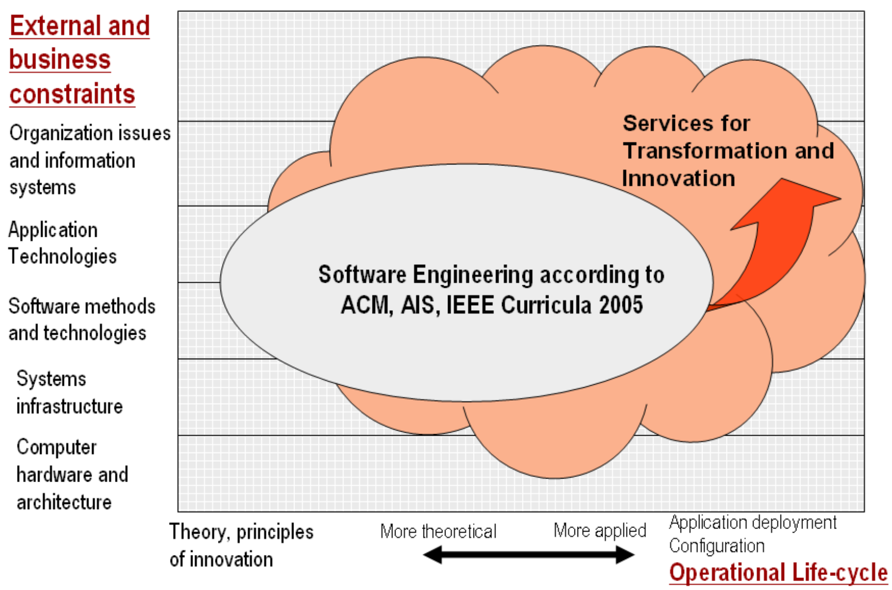
[1] This observation is based on anecdotal evidence gathered from discussions with our industry partners. In other words, even where the organization had the domain expertise, external consultants were relied upon to lead and deliver the project.

[2] The role of Software and Services within CORDIS – European FW …In the Information Society we have several raw materials. They come in form of commodity components, e.g. computer chips or software components. Software is more than meets the eye. In value and employment, software services like running a network for inter bank balances, and embedded systems are much bigger than office suites. A luxury car today has 50-60 processors which all need to be "filled" with software (environment-, security-, navigation requirements). New services emerging from the convergence of Telecom networks, media and the IT

This shift is clearly reflected in the career and hiring trends around students graduating with an undergraduate degree in the computing fields. Evidence (primarily gathered through exit surveys) have indicated that nearly 70% of the computer science students at The Ohio State University (OSU) begin a professional career after their undergraduate degree (as opposed to going on to graduate school), and that the majority of these students are hired by internal departments and external system integrators that provide IT-enablement to the service industry.[3]

This trend points out the need for a more applied, integrated, interdisciplinary education. This need has also recently been recognized by both academia and industry. Within academia, SE education as defined in Computing Curricula 2005 (a joint effort of ACM, AIS and IEEE) is as illustrated in Figure . Here the shift in knowledge needs has been recognized – the field of SE (as reflected by the oval) has been expanded in this version to incorporate *applications and information systems* vertically and to become more *applied* horizontally.

**Figure 1: The oval reflects Software Engineering as defined in Computing Curricula 2005. The outer cloud reflects the expanded scope of SE that also considers business constraints on IT operations.**




However, the Computing Curricula 2005 does not identify the details of where the software engineering curriculum falls short in application, and how this curriculum needs to be evolved and enhanced. Thus, in Figure , we illustrate what we hypothesize as the critical areas of curriculum development needed as the enlarged Services for Transformation and Innovation[4] cloud.

---

industry provide an important opportunity for increased European competitiveness. The real challenge is to get new value out of the ability for software and information to travel on networks. Europe is strong in this area. Its key ability is engineering of software and services. This needs critical mass also in times of convergence. We therefore develop the ingredients one needs Europe-wide to provide innovative products and services. We develop the software supply industry but also look at the human capital needed for it and help with the Technology Transfer from the supplier to the user.

[3] This trend is not limited to undergraduate students alone; a significant percentage of our graduate students are now also hired within technology *using* companies and their system integrators.

[4] The concepts of transformation and innovation used here are as well described by Denning (http://www.acm.org/ubiquity/interviews/v5i8_denning.html) as follows: "Innovation is the transformation of practice. In this definition, community can be small, as in a workgroup; or large as in the whole world. A transformation of practice in the community won't happen unless the new practice generates more value

Our understanding of the Transformation and Innovation in services that motivate our long-term curriculum-mining effort is based on input from thirty-five medium-to-large organizations (including several multinationals) in an enterprise architecture community in the mid-western United States. Our use of the term 'innovation' encompasses operational aspects of complex systems that have to function in the business context. Finally, in the articulation of Figure the reasons for the shift from SE (as described in CC 2005) to Services for Transformation and Innovation *are not widely understood by academia*. For example, the Software Engineering Body of Knowledge [SWEBOK04] does not include software *integration* as a subject area. Technology developers do understand this shift better. For example, companies like IBM, Microsoft and HP make available a large body of application knowledge that is not covered in curricula (see Figure 2 for topics identified by IBM). Additionally, standards bodies and associations like OMG (http://www.omg.org), W3C (http://www.w3c.org), ISO20000 [ITIL], Capacity Management Group [CMG], TOGAF [TOGAF], and the Software Engineering Institute (http://www.sei.cmu.edu) are just a few that have established themselves by creating practice knowledge outside of academia. In a field like SE, much of this knowledge is vital for the successful deployment of complex systems made up of existing and emerging technologies.

Before we develop curriculum materials that are adequate to communicate the appropriate knowledge, further understanding underlying trends is key. We describe these trends next.

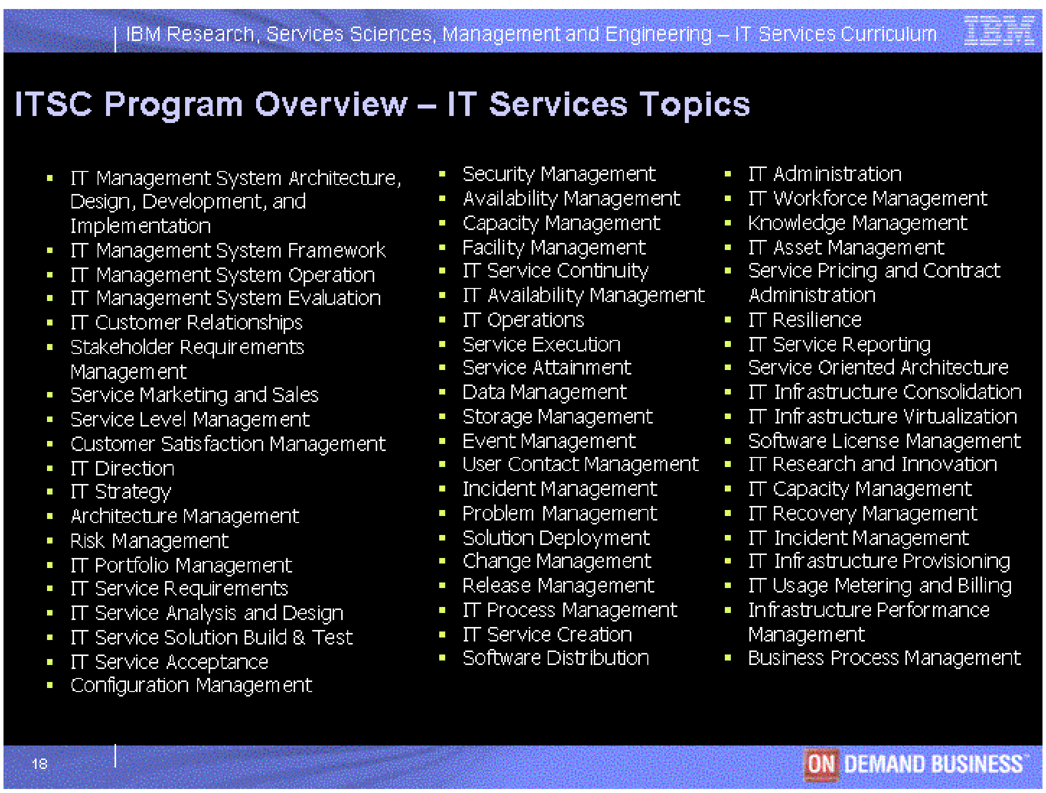



**Figure 2: Topics in IT Service Management (courtesy IBM SSME)**

to the members than the old ..."

### 2.2. Trend: Service-Oriented Businesses

With the growth of "on-demand" businesses, a major area where SE must evolve is in support of services. The study of the field of services has come to be known as "services science." As Dr. Jim Spohrer, Director of IBM Almaden Services Research Center writes [IBMSSME], "Service Science is really trying to address the need to get more systematic about innovation…how does technology change, how does the business model change and how does the organization change? How does all that work? And just to make it even more realistic, how does the client's demand change with these transformations?" These questions translate to Business-IT questions, such as:

- How can the business sense, and respond to, external shifts using IT?
- How do we connect business goals of transformation and innovation to IT policies?
- How do we structure, architect and compose services consisting of automated services provided by systems and manual services provided by organizations?
- How do we model services and reason about their improvement when services include both automated IT and manual services?
- How can organizational design and engineering be integrated into software engineering?
- How do we teach the application of advanced computer science technologies, such as data mining, machine learning, pervasive computing, and distributed computing to enterprise systems that deliver services?
- What really are the interdisciplinary aspects of services? Is a services curriculum best taught only through interdisciplinary group-based projects?
- What do we need to borrow from business and the social and behavioral sciences to concretely teach about the human element in software engineering (given the fact that almost every enterprise-scale information technology project is actually an enterprise *transformation* project that impacts the people in the organization)?

### 2.3. Trend: Complex Systems Evolved Through Integration

Rarely are today's systems built from scratch. The post Y2K and .COM era has also propelled to the forefront the issues of enterprise-wide information flow, interoperability and integration of IT components into systems that *evolve* to become extremely complex, as opposed to the complexity being designed in[5]. The resulting need is for knowledge to manage complex systems and their life cycles.

However, SE that is taught in the classrooms and the needs of the IT users in industry has significantly diverged over the last two decades. For example, much of our education is primarily targeted at designing and building architectures, systems and applications *from scratch.* While requirements' gathering is introduced, the effective capture of requirements originating from the context of existing system requires business analysis skills that are not introduced. "Creational" tools and technologies such as programming languages, compilers and integrated development environments that primarily support coding are well-covered. However, the largest segment of the technology industry, specifically the technology consuming industry (comprising all those organizations that use information technology to enable their actual business - such as banks, airlines, healthcare and retail), spends most of its effort on systems *integration* using integration technologies and managing the *evolution* of architectures. Dealing with the *deployment*

[5] As mentioned earlier, this has been accompanied by increasing numbers of Software Engineers hired into the IT *using* industry.

and *management* of these complex, evolved systems on complex infrastructures poses additional types of questions to be addressed:

- How do we integrate infrastructure life-cycle management, operations and, specifically, the IT Infrastructure Library [ITIL] methodology standard into the SE curriculum?
- How does a curriculum around Model-Driven and Product-Line Architecture [SEIPLA] include integration and legacy environments?
- What specific tools and technologies around enterprise integration and technology-enabled services are sufficiently established for incorporation into our curriculum? Examples of such technologies include workflow systems, Enterprise Service Bus products, message-oriented middleware and infrastructure planning and monitoring products.

### 2.4. Trend: Global and Remote Development and Deployment

Picking the right strategy for global development and deployment is critical in today's economy. This raises additional issues that must be addressed:

- How must teams address "chunking" [Mockus01] the work and distributing it based on functionality, localization or life-cycle stages [3]?
- How can a system be architected to facilitate its distributed development and deployment? Just as a monolithic piece of software is not amenable to distribution during development, so is a system with components that exhibit a high degree of coupling. While the introduction of service orientation has helped tremendously system decomposition is still key. Another angle to deal with here is tools needed for development and deployment that can help with data availability, collaborative editing, change control, and monitoring.
- How can the impact of non-technical issues dealing with cultural and social differences between geographic locations be communicated and addressed? These issues considerably complicate a distributed development effort. A pleasant working relationship between members of a team is a must if progress is to be guaranteed. We cannot underestimate the effects of understanding and adapting to the different cultural makeup of team members.
- How can life-cycle processes such as knowledge transfer, quality assurance and configuration management be addressed in a global context? Knowledge here will take many forms including application knowledge, development standards, technology knowledge and company culture knowledge. Knowledge Management technology can considerably ease the pain of sharing information but must be deployed early and broadly.
- How can the factors of language and time figure into the communications challenge? Especially in projects that may need a language specific user interface. Differing time zones may at first glance appear inconsequential but temporal dispersion does add to logistical challenges such as arranging for knowledge transfer or videoconferences across teams.

These issues clearly point out the need to specifically consider the effects of globalization on process and product aspects of SE.

## 3. Practice Pedagogy - Mining SE Knowledge Assets

### 3.1. Motivation and Objectives of our Work

Our motivation is to 1) reduce the multi-decade[6] time lag between curriculum and practice and 2) increase industry relevance without compromising quality. Industry

[6] For example, consider the time lag between use of relational data bases in industry and its introduction into classes; the widespread identification of ITIL in mid 90's by industry, especially in Europe, and its adoption into academia still in infancy.

relevance here means many things – timeliness, high impact, principles of problem solving, field performance etc.

In summary, along with other researchers [Rodriguez, IBMSSME], we observe that new knowledge assets have to be created to address:

- Evolution and integration of IT systems in support of business services rather than construction from scratch and software products.
- A holistic life-cycle approach targeted to understanding and achieving business objectives, rather than primarily on technical aspects and the *functional software development* process alone (for example, addressing critical requirements such as disaster recovery).
- Global collaborative development models rather than co-located development only.

Note that the creation of curriculum content is not simply a matter of collating existing literature from academic and practice publications, firstly because there is insufficient room to do this within our educational programs, but more importantly because we have to teach how the knowledge must be *applied* in context.

Also, this type of knowledge is not well reflected in the existing curriculum. One reason is that research undertaken by academia is on hard but not necessarily on high-impact problems faced by industry. Academia typically focuses on problems that are "suitable" to be researched in the traditional manner leading to funding and publication in journals. And since industry funding is usually tied to high-impact and immediate problems with immediate deliverables, faculty usually stays away from this type of research. An example of this is program analysis, an area where academic SE research energy has been focused, and where graduate and undergraduate courses are now being developed. Except in very limited way, this area of research has yielded little in terms of knowledge applicable to "where the industry action is" – i.e. in integrating systems for service delivery. This, we posit that we do not quite know what content to teach, what stage of the student's evolution to teach it in, or how to make the necessary academic and institutional changes to impart the needed more knowledge effectively.

**Our objectives within this paper are thus:**

- Develop a pedagogical framework (illustrated in Figure 3) for identifying, vetting, and delivering practice-relevant knowledge within an academic setting.
- Target the area of framework application – i.e. Service Transformation and Innovation - to mine curriculum assets.
- Demonstrate the concrete successes of this practice framework in terms of a) new Knowledge Frameworks for the SE curriculum, b) best practices enhancements to industry, c) timeframes, and d) integrated, interdisciplinary *research* themes around enterprise-scale *application* of technology.

Our experience is based on twenty-five field case studies of Industry-University collaboration at The Ohio State University[7] and the Indian Institute of Technology[8], Bombay.

[7] The Ohio State University and CETI - http://ceti.cse.ohio-state.edu/ceti
[8] IIT Bombay – http://www.iitb.ac.in

### 3.1 Background in Learning Research

The framework for developing SE assets builds upon available methods for learning. For each of the steps – i.e. analysis, mining, vetting and delivery introduced in Figure 3 – we introduce background details next:

- **Task-conscious or acquisition learning** [Rogers03]: Acquisition learning is seen as going on all the time. It is 'concrete, immediate and confined to a specific activity; it is not concerned with general principles'. Here, whilst the learners may not be conscious of learning, they are aware of the specific task in hand. Capstone courses, internships and professional practice itself contributes to this aspect of practitioner education.[9]
- **Learning-conscious or formalized learning** [Rogers03]: Formalized learning arises from the process of facilitating learning. It is "educative learning' rather than the accumulation of experience. To this extent there is a consciousness of learning - people are aware that the task they are engaged in entails learning. Learning itself is the task. Formalized knowledge makes learning more conscious in order to enhance and accelerate it. The pedagogical objective is to mine and organize knowledge in order to create formalized learning tools that will *accelerate practitioner learning*.
- **Double-loop learning** [Argyris78]: This form of learning involves the detection and correction of error. Where something goes wrong, it is suggested, an initial port of call for many people is to look for another strategy that will address and work within the governing variables. In other words, given our chosen goals, values, plans and rules are operationalized rather than questioned. This is single-loop learning. An alternative response is to question to governing variables themselves, to subject them to critical scrutiny. This is double-loop learning. Such learning may then lead to an alteration in the governing variables and, thus, a shift in the way in which strategies and consequences are framed.
- **Reflective practice** [Schon83, Schon87]: Reflective practice is a specialization and refinement of double-loop learning. The capacity to reflect on action so as to engage in a process of *continuous learning* is one of the defining characteristics of professional practice. The cultivation of the capacity to reflect *in* action (while doing something) and *on* action (after completion of the action) has become an important feature of professional *training and delivery* programs in many disciplines, and its encouragement is seen as a particularly important aspect of the role of the mentor of the beginning professional.

Our approach here is to convert task-conscious learning to formalized-learning through the use of reflective practice. Building on what works in "reflective practice" [Schon83, Schon87], an important point has been made that practitioners must themselves develop *frameworks for complex problem* solving and continuous learning. We call this a 'Knowledge Framework' and discuss that next.

### 3.2. Knowledge Frameworks

For the reflective community to practice and succeed with double-loop learning, there must be plans to guide actions and defining desired results, ways of monitoring the field of action, means of tracking and recording results achieved and comparing what is observed against what is desired, and set of components for taking action when there is a difference large enough to care about between the two. Essentially, these ways, means and records are *frameworks*. These frameworks can be based on a variety of representational tools – patterns [Gamma95], best practice processes and templates,

[9] From a pedagogical perspective, acquisition learning provides a context for a reflective practitioner to characterize the needed knowledge assets for the problem at hand and the missing knowledge to be mined.

ontologies [Chandrasekharan99], and so on.

To achieve improvement, the community must create and evolve representations, principles, and analysis we call *Knowledge Frameworks,* which are capable of representing and guiding practice and measuring performance ([Denning 2003]). These frameworks must go *beyond* examples and case studies such as those used in the business curriculum [Barnes 94]. While the case study method is invaluable for after-the-fact understanding of the decision-making process, it is inadequate for systems analysis, continuous improvement, and pro-active decision-making that is involved in IT practice.

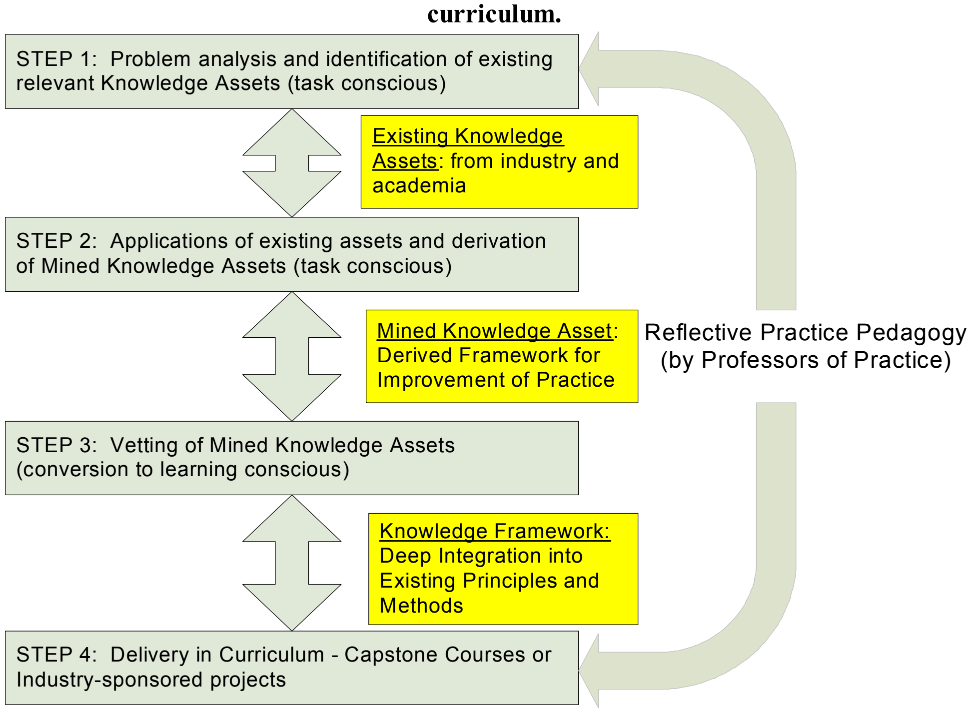
As with any complex design method involving large solution spaces, the frameworks must be able to:

- Identify and characterize the precise business problem
- Analyze the context, including the existing knowledge assets that it enhances or replaces
- Surface all of the relationships and behaviors (performance etc.) that must be optimized during the lifecycle of the systems and organizations involved
- Apply *existing or enhanced* principles and methods for analysis for solving the problem and creating the solution.

We view reflective practice as being essential to the *vetting* of new knowledge and to the creation of *Knowledge Frameworks*. Once this process is completed, we say that the resulting Knowledge Framework is *deeply* integrated into the existing curriculum.

### 3.3. Practice-Relevant Pedagogy Approach

**Figure 3: Reflective Practice Pedagogy: Process, Knowledge Asset and Framework Creation for SE curriculum.**




Having motivated the need for practice-oriented pedagogy (both for teaching and related content development), and presented background information, we now present our *pedagogical framework* to *mine Knowledge Assets, and vet and deliver Knowledge Frameworks for SE.* Our approach includes the following steps (overview in Fig):

**Step 1: Problem analysis and identification of existing Knowledge Assets:** During the project engagement process, we typically began by identifying a) solution needs, b) available materials - knowledge assets in academia, practice and vendor communities - that could apply to the project, and b) learning outcomes. As we proceeded through the project, we monitored these learning outcomes and re-evaluated them as needed.

- **Work products of STEP 1 - Existing Knowledge Assets:** The output of an industry-sponsored problem analysis phase is an identification of the Knowledge Assets needed in developing types of solutions and the constraints that apply on those solutions.

**Step 2: Application of Existing Assets and Mining of New Assets:** During the project execution missing knowledge gaps and assets are identified.

- **Work product of STEP 2 - Mined Knowledge Assets:** Mined assets are likely to be interdisciplinary - the disciplines of (technology-oriented) Computer Science, (applications-oriented) Information Systems, Systems Engineering and Business are not connected within academia, but are in practice. For example, new knowledge might help answer: How would new information delivery requirements drive the total cost of ownership or return on investment of the enhanced operational system?

**Step 3: Vetting of Mined Knowledge:** We use the term *vetting* to connote not only the validity of knowledge at a point in time, but also the development of principles and ongoing validity through continuous improvement. From a pedagogical perspective, it is important to not identify and introduce into University curriculum that, a) still does not genuinely meet practitioner needs or, b) even if initially industry-relevant, quickly becomes obsolete because of changes in the field or, is c) better delivered through other means (such as professional development workshops and seminars). Essentially, it is important to carefully make the case that the currently accepted curriculum - that has arguably been established through decades of careful academic oversight, even if not through genuine industry validation - must be reworked. This knowledge mining step is therefore important to ensure that the newly identified curriculum content is not, for example, simply the "warmed-over" practices of a couple of major industry players delivered within a university setting[10].

- **Work product of STEP 3 - Knowledge Framework:** At the end of the project, participants were required to write reports as well as research papers from their experiences. These reports as well as faculty and industry member experiences were used to identify and extract curriculum material. The eventual result of this step is, therefore, a Knowledge Framework providing the structure for principles, analysis and continuous improvement. We use the term 'deep' to connote interdisciplinary principles and methods that have to be integrated to yield measurably more realistic and successful practice solutions [Denning 2003].

**Step 4: Delivery in SE Curriculum:** This delivery requires the effective management of the process Practice Pedagogy. It is well accepted that practice knowledge is best delivered in the context of Capstone projects, industry-sponsored projects or Masters' theses.

[10] At the same time, it is important to ensure that academia keep current with vendor-provided technology. Fielded enterprise systems, for example Service Management and IT Configuration Monitoring, often outstrip what is taught.

### 3.4. Critical Success Factor: The Professor of Practice Role

We define the *Practice Professor* role as a specialized faculty role that identifies, vets and delivers the Knowledge Assets and Frameworks in a double loop learning process. Academia is becoming increasingly sensitive to the challenge of teaching advanced practice skills [MEAD00] but it lacks faculty in the capacity of Professor of Practice. Some quick-fix approaches taken are the hiring of industry professionals on a part-time basis [MEAD00] or have academic faculty teach practice courses. These approaches cannot have long-term success for many reasons:

- The quality of the pedagogy is inconsistent - the focus of most professionals is their job and not the development of their skills to meeting teaching objectives,
- The knowledge held by non-academic practitioners is often limited by what they have actually experienced in their particular career,
- Compensation schemes within the universities cannot attract high-value industry consultants that survive on new knowledge consolidation, and
- The *task conscious* methods require the embedding of academic faculty within the industry environment. To develop knowledge suitable for formalized teaching, we must also require the embedded investigators to be involved in "double-loop" learning and "reflective practice" within the learning communities of practice. There are very few established and accepted processes for faculty skills renewal, with extended time periods that are long enough (five or so years) for experiencing high-impact issues related to a few complete project life-cycles. (The short-term programs available within foundations like the National Science Foundation (NSF) are not long enough to experience the cradle-to-grave aspects of complex project cycles).

We conclude this section by stating that we propose our Practice Pedagogy approach with some confidence of success. Firstly, industry is aware of and is responding to the need for industry-university collaboration and faculty embedding. One of them is the Boeing Welliver Faculty Fellowship Program[11]. Secondly, there are industry bodies such as TOGAF [TOGAF], ITIL [ITIL] and, certainly, the Software Engineering Institute[12] where several elements of our approach are being currently practiced. Finally, our interactions with industry to date have helped us identify and mine valuable practice knowledge required for our students.

Next, we present examples and the range of different consequences resulting from our practice pedagogy experience.

## 4. Examples of Mined Knowledge for Transformation and Innovation

This section describes examples of curriculum material mined through practice from two categories of projects conducted over a four-year period. The two categories are as follows:

- Capstone courses at OSU (CSE 758 SE Project) and IITB (IT 607 Software Engineering). Within these courses, we have to-date conducted over twenty-five student projects sponsored by local and national industry, non-profits and

[11] http://www.boeing.com/companyoffices/pwu/fellowship/objective.html
[12] http://www.sei.cmu.edu

entrepreneurs. Projects have ranged from simple dynamic web sites, to components of enterprise systems to mobile PDA and cell-phone applications. All the projects were specifically targeted towards solutions and improvements within the industry context.

- Industry and university research projects. These projects have been a mix of projects in (a) technology strategy (b) pattern-mining (c) architecture evaluation (d) large-scale collaborative research project management and (e) professional development workshops, and have been undertaken by teams composed of faculty, graduate and undergraduate students *and* industry practitioners.

The examples have been selected with a view to illustrating the mined assets in the theme of *"Service Innovation and Transformation"* (Figure 1). In each example described we present:

- *Project* - a brief description
- *Problem and analysis* - the problem to be addressed and existing knowledge assets,
- *Learning outcome* – the desired outcome, providing insight on what is not currently learned with existing assets.
- *Consequences* - mined knowledge assets addressing the gap between "the known and the unknown", the resulting integrated knowledge framework, impact to the organization, and finally impact and enhancements to curriculum.

**Project:** Information Technology Strategic Plan for the City of Columbus

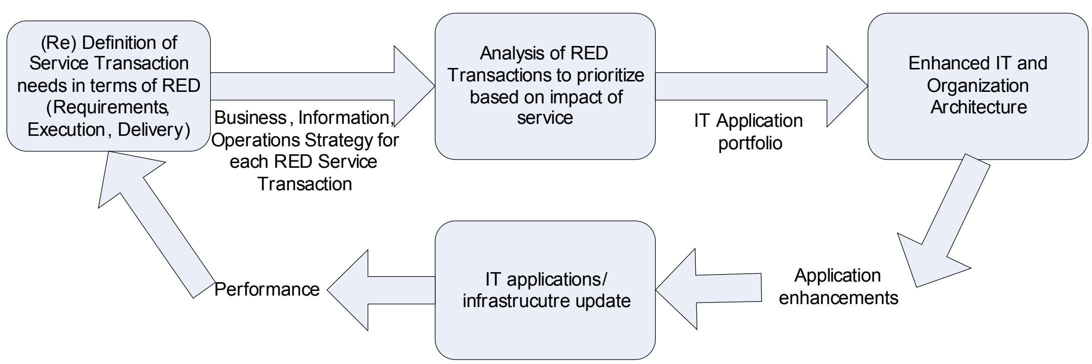



**Figure 4a: Process Framework for Information Technology Strategic Planning**

**IT Application Portfolio - Missing Applications**

| ***City Department*** | Case Management | Performance Management | Document Management | Asset Management | Forms Engine | Payment Engine | Project Management | Internet/Intranet | Wireless | GIS |
|---|---|---|---|---|---|---|---|---|---|---|
| *Development* | X | | X | | X | | | X | X | X |
| *Utilities* | X | X | X | X | | X | X | X | | |
| *Public Service* | X | | X | X | | X | X | X | X | X |
| *Public Safety* | X | X | | X | | X | X | X | X | X |

**Figure 4b: Application Portfolio – Missing IT Applications and Infrastructure**

**Problem and analysis:** Create a road map that aligns Information Technology with City service mission, goals and objectives. Existing assets included IT strategic planning guides [Cassidy98], business strategy guides [Porter79] and the TOGAF [TOGAF] enterprise architecture framework.

**Learning Outcome:** Learn to address the central question behind IT strategic planning: "How can an implementation roadmap be developed for a very diverse collection of IT organizations, systems and services (such as those provided by a city or a hospital), aligned to business service objectives".

**Consequences:**

*Mined Knowledge Assets:* We determined that existing assets in Enterprise Architectures are not prescriptive. Two assets were consequently developed: (a) a process framework (shown in Fig 4(a)) was developed from existing work described above and from our prior research in decision-making in complex adaptive architectures [Ramanathan05] and (b) a knowledge capture framework consisting of an application portfolio template and shown in Fig 4(b).

Using these new assets, the strategic plan was developed consisting of two parts (a) a roadmap for business-IT alignment centered on strong account management and (b) prioritized IT initiatives, with the primary one being a central 311 system for citizen access to non-emergency City services. The project time frame was two years.

*Knowledge Framework:* The process of developing a knowledge framework has begun with initial results published in a CACM special issue on Complex Adaptive Systems [Ramnath05]. These projects have enabled research on extending TOGAF and ITIL with the RED representation [Ramanathan05] along with analytic methods for architecture evaluation and activity-based charge-back models.

*Impact to the organization:* The strategic plan continues to be the basis for the City's IT operations 3 years after its creation. The strategic plan also spawned additional follow-on projects such as an ATAM-like [Bass03] evaluation of the technical architecture of the 311 system, and the development of an activity-based charge-back model for technology services.

*Enhancements to Curriculum:* The above frameworks have been incorporated into a new technology management track of our Professional Masters' curriculum, currently under development.

**Project:** Software Engineering Capstone (CSE 758) at the Ohio State University.

**Problem and analysis:** In this 1-quarter course, student teams do industry-sponsored projects, while following professional practices in software development. Assets included textbooks on planned [IBM97] and agile [Beck04] methodologies and material on integrating planned and agile methods [Boehm04]. We needed to establish the use of software engineering methodologies within this course. Planned methodologies were seen primarily as after-the-fact documentation techniques, and students attempting to apply agile methodologies simply gravitated to ad-hoc methods.

**Learning outcome** – Demonstrate the value of SE methodologies though successful project outcomes.

**Consequences**

*Mined Knowledge Assets:* This consisted of the process selection framework from [Boehm04], a catalog of planned and agile work-products drawn from [IBM97] and [Beck04] and a project plan template reflected in the expected course milestones. Students were expected (a) to create a subsystem model of their system (b) to identify the subsystems that were better suited for either planned or agile development, and then (c) identify the work-products from a catalog of agile and planned work-products to capture the design and implementation.

*Impact to the Curriculum:* In this case while the knowledge assets and the framework existed, the mined insight was in the integration into the course. For example, in the project-plan template, 2 checkpoints were key. These consisted of an early project checkpoint where high-level requirements, a high-level architecture and the selected work products were presented and a mid-term check-point where the teams were expected to demonstrate "process maturity", i.e. for both completed and new requirements they were expected to be able to clearly identify the work-products they would use to analyze, design and implement the requirements.

The acceptance of software methodologies was significantly improved. Students explicitly tailored requirements, analysis and design work-products to the project shape before progressing through to implementation. This tailoring process brought buy-in, which resulted in active use of the methodology. One example of genuine buy-in was that students could clearly articulate traceability relationships between the work-products.

These frameworks are now established course resources for the Software Engineering Capstone course (CSE 758) and its pre-requisite Fundamentals of SE course (CSE 757).

**Project:** The Extreme Scale (ExScal) Sensor Networking Project[13]

**Problem:** There were two problems as follows:

- High-level architecture definition for the project was done in an ad-hoc manner. An appropriate characterization of the application was needed and alignment shown between the application characteristics and the selected architectural components.
- Since this was an exploratory research project, expectations of the sponsor were undefined, and the potential capabilities of the delivered system were unknown. Detailed project planning was difficult to accomplish, yet this was a highly visible project that needed to be transparently managed.

From an existing asset perspective, there existed a literature survey of 14 widely-varying pervasive and mobile computing applications, and pervasive computing middleware architectures. Also there were Agile project management guides [Highsmith04].

**Learning outcomes:**

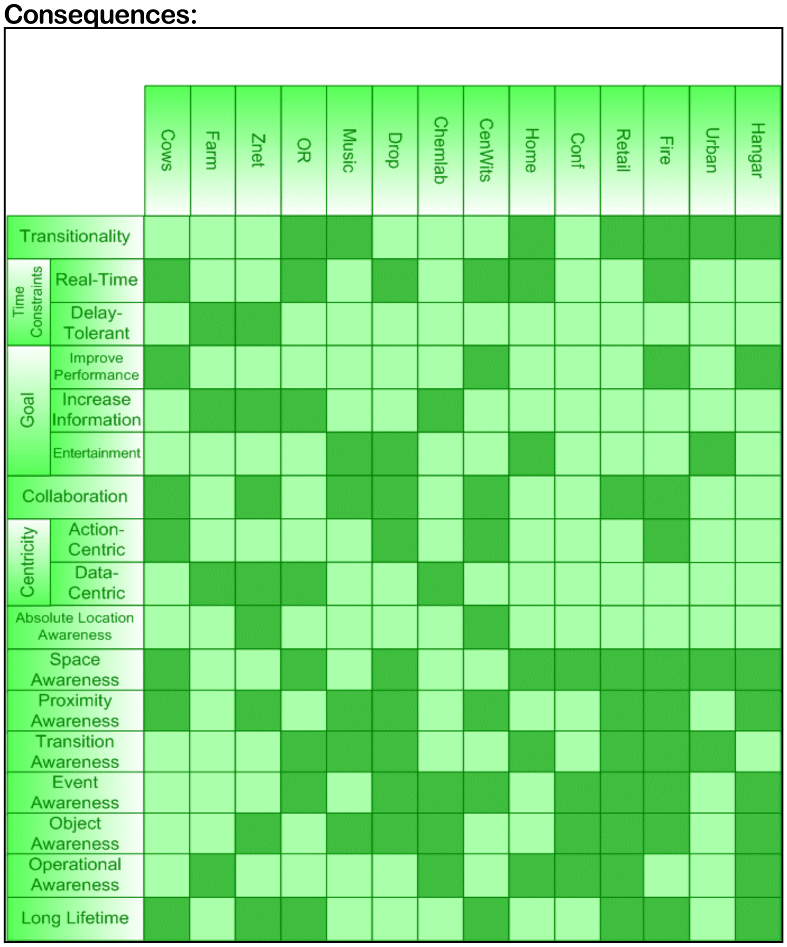
- Develop the capability to rapidly assess architectural alternatives for pervasive

[13] Funded by the Defense Advanced Research Projects Agency (DARPA) [Exscal04, Arora05]

computing applications.

- Develop the capability to manage large-scale, open-ended research projects.

**Consequences:**




*Mined assets:* A "transcribe and converge" [IBM97] based taxonomy development process. A useful and complete taxonomy was developed [Dombroviak06]. The taxonomy immediately helped identify the broad components necessary to support ExScal and other pervasive computing applications (such as localization, time synchronization, classification, sensing, and so on). The taxonomy was also used to define the architecture of an application-level pervasive computing simulator, the design and implementation of which has been reported in [Dombroviak07].

**Figure 5: Using this chart, we can compare the characteristics associated with each application. This allows us to quickly compare and contrast the requirements of applications. It also shows that each of our example applications, which we know differ, is categorized differently by our taxonomy.**

*Knowledge Framework:* There were both immediate and longer-term implications of the taxonomy. For example, by assisting in defining the components of the application architecture, the taxonomy also helped us define the project plan (Fig 6). Essentially the project plan became a set of milestones, where, starting from a baseline set of capabilities, each milestone was defined in terms of increasing capability of each of the components. Thus the taxonomy provided a project planning discipline for bringing transparency to projects with extremely open-ended requirements.

*Enhancement to Curriculum:* The taxonomy (Fig 5) and an excerpt from the related project plan (Fig 6) are now course resources in the Fundamentals of Software Engineering course (CSE 757).

| | Requirement / Metric | EXT | | |
|---|---|---|---|---|
| | | Integration 0 | Integration #1 | Integration #2 |
| DEMO INFORMATION | Goals and Objectives | New XSM Mote integration. Integration with 2 Stargates. Localization. Acoustic-based detection. Infra-red based detection. Integration w/routing at Tier-1. Confidence in Tier-1 routing. | Confidence in mote. Confidence in enclosure. Integration with partner components (localization, time-sync, routing, reprogramming). Tier 2 networking - small topology w/5 hops. | Variability across sections. Fault injection. Confidence in acoustics--150 Stargates. Simulation results of application at scale of 1500 XSM. |
| | Date | April 30, 2004 | July 30th | August 30th |
| | Location | Columbus, OH | Columbus, OH | Columbus, OH |
| SENSOR NETWORK REQUIREMENTS | Coverage Area | 180m x 36m | | |
| | Network Size | Tier 1: 100 (XSM); 100 (MICA2)<br>Tier 2: 5 (Stargate)<br>Base Station: 1<br>Other: ??? | | ~~Tier 1: 500 (XSM)~~<br>Tier 2: 150 (Stargate)<br>Base Station: 1<br>Other: Tier 3 relay |
| | | Integration 0 | Integration #1 | Integration #2 |
| | Goals and Objectives | New XSM Mote integration. Integration with 2 Stargates. Localization. Acoustic-based detection. Infra-red based detection. Integration w/routing at Tier-1. Confidence in Tier-1 routing. | Confidence in mote. Confidence in enclosure. Integration with partner components (localization, time-sync, routing, reprogramming). Tier 2 networking - small topology w/5 hops. | Variability across sections. Fault injection. Confidence in acoustics--150 Stargates. Simulation results of application at scale of 1500 XSM. |
| | | | | |
| | Date<br>Location<br>Coverage Area<br><br>Network Size | April 30, 2004<br>Columbus, OH<br>180m x 36m<br><br>Tier 1: 100 (XSM); 100 (MICA2)<br>Tier 2: 5 (Stargate)<br>Base Station: 1<br>Other: ??? | July 30th<br>Columbus, OH | August 30th<br>Columbus, OH<br><br><br>~~Tier 1: 500 (XSM)~~<br>Tier 2: 150 (Stargate)<br>Base Station: 1<br>Other: Tier 3 relay |

**Fig 6: ExScal Milestone Based Project Plan. Columns show the milestones. Rows show the attributes of the milestones, while the cells show the values for each of the attributes. Milestone attributes were derived from the taxonomy shown in the previous figure.**

<u>**Project:** Improvement of the PC-fulfillment process at the OSU Medical Center.</u>

**Problem and analysis:** Existing many-to-many relationships between Business and IT services are complex to manage and challenge the agility of an enterprise. While many best practices for services support and delivery existed [George02, Hammer93, Haeckel99, ITIL, Jones 03], they were limited in terms of *prescriptive* methods for IT *service improvement*. In addition, existing improvement methods were not designed for aligning IT services with improved business performance (such as achieving Lean use of resources).

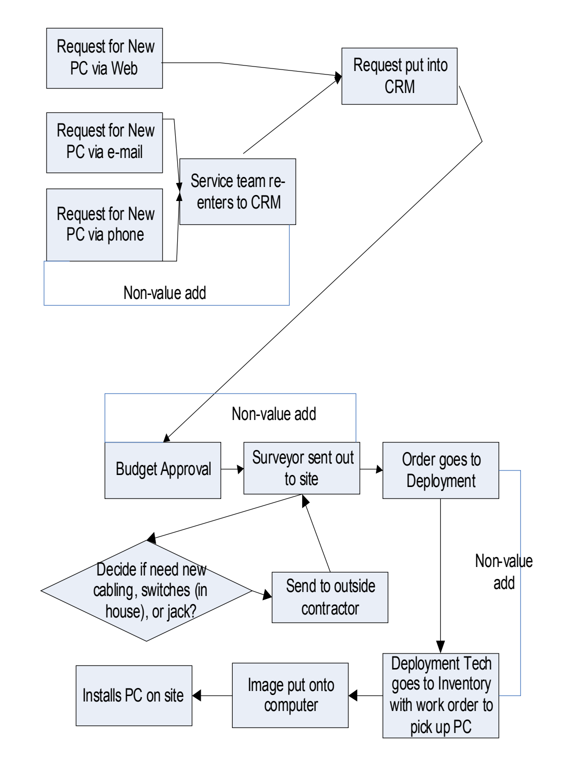
**Learning outcome:** Address the central question behind IT strategic planning: How can an *implementation* roadmap be developed for a very diverse collection of IT organizations, systems and services (such as those provided by a city or a hospital), aligned to business service level objectives and Lean?

**Consequences:**

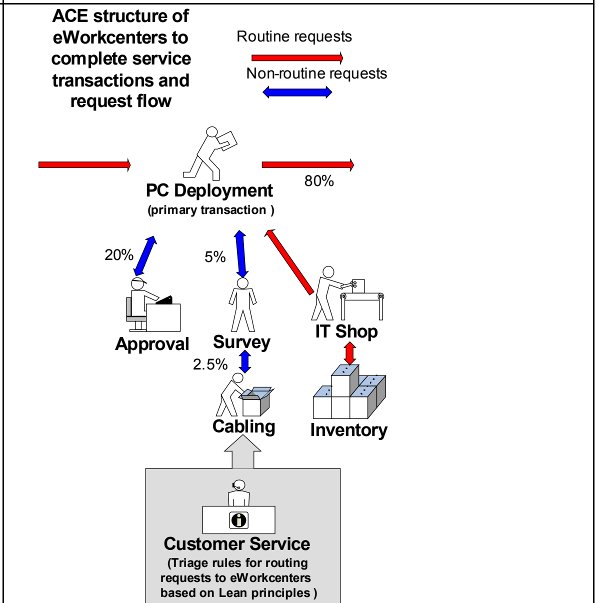
*Mined Knowledge Assets:* The Adaptive Complex Enterprise framework [Ramanathan04] of transactions for improving the deliverables of a complex organization (humans, processes, software systems, assets etc.) is illustrated in Figure 7b. This representation scheme (underlying the As-is in Figure 7a) captures request types, routings, flow rates to eWorkcenters for processing and mimics factory floor work centers. This virtual To-be structure also permits the application of the Lean methods by virtually grouping shared services into eWorkcenters to which only non-routine requests are routed.

*Knowledge Framework*: A complete case study in IT for the healthcare industry is now available to illustrate and become a starting point for a widely applicable Lean method for shared IT services [Ramanathan 07]. Such a framework would enhance software architectures by providing principles for decision-making and an engineering approach to the improvement of complex systems. Future industry relevant research in the area of simulating the interactions between the *global* system and its *local* autonomous (self-managed) entities for management of complex systems has been identified.

**a) As-is Process with non-value-add steps due to the processing of both routine and non-routine steps in the same manner.**




**b) To-be Framework of eWorkcenters and transactions to complete requests**




The framework develops a roadmap (tasks and priorities and impact to business) to reduce the time to install new PCs from 17 days to 1 day.

The method application also defines business service levels and relates them to operating levels.

**Figure 7: Adaptive Complex Enterprise Structure of Transactions and Virtual eWorkCenters providing IT and Human services and related analysis for improvement.**

*Impact to organization:* Lean decision-making principles provide operating level agreements to accomplish service level agreements to meet business performance objectives.

*Enhancement to Curriculum:* We now have examples of how the Lean methods from systems engineering can be applied to IT shared services improvement and provide precise system improvement methods to complement the ITIL best practice. This is being piloted for senior or graduate-level students.

**Project:** Enterprise Architecture Pattern Mining at an International Publishing Company

**Problem and analysis:** The problem was to look across the enterprise and

- o Identify and reduce *redundant* IT investments

- o Consistently ***engineer*** IT systems and decrease *project delivery* times
- o Define a *future-state* architecture driven by business needs

While many projects had extensive functional documents, most projects had very poor non-functional and architecture documentation. Most of the existing work in architecture patterns ([TOGAF], [Fowler02]) is focused on functional rather than on fine-grain services or non-functional aspects. Additionally, best practice literature with vendor *independent* architecture information was very scarce. With the help of team members we attempted to re-construct architecture-related experiences as architecture patterns. In collaboration with the project manager, different project contacts were identified and the pattern summaries were developed and discussed collaboratively with the teams before aligning the available content to the summary for the final pattern document and review.

**Learning outcome:** Have an understanding of a comprehensive range of architectural patterns for solving interoperability problems between enterprise systems encountered in the field.

**Consequences:**

*Mined Knowledge Assets:* The framework for increasing reuse began with understanding the role of individual patterns in the context of other patterns. This was accomplished by a dimensional framework that views the functions of the enterprise as an internal value-chain as illustrated on the *left* of the Figure 8. The mined patterns and their detailed characteristics and inter-relationships could now be better understood as on the *right*.

*Knowledge Framework:* The architecture mining lead to a more complete framework of patterns than in TOGAF. It also identified opportunities for model-driven life-cycle management (not just creation, but also monitoring and continuous enhancements).

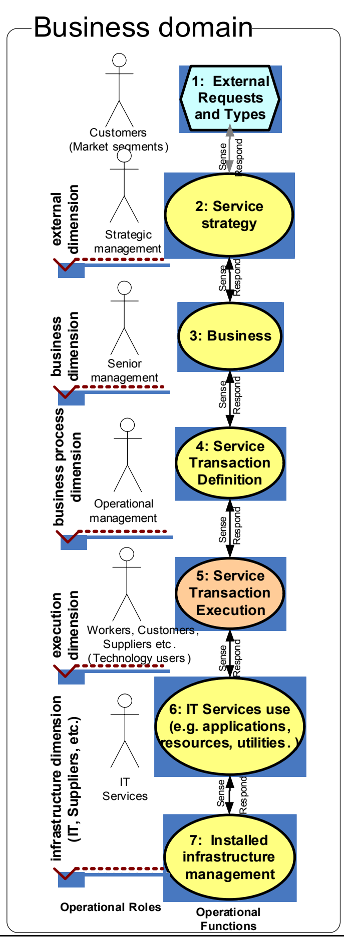
*Impact to the organization*: Pattern-based insights within the organization included:

- ° Identification of where market trends and opportunities for new services are impacting application architectures (this ensures competitiveness). For example, several projects are developing more flexible ways of managing customer communities as a way of delivering custom services.
- ° Identification of frequently used application (functional) patterns that can be used to develop new services rapidly, test these with challenging requirements and proof of concept projects.
- o Identification of opportunities for reuse and standardization across the corporation. Several projects are developing better methods for using enterprise-service-bus-based architectures.
- o Prioritize patterns for standardization based on high occurrence and high impact to business processes. For example, several projects are looking at security, trust and rights issues. Prioritize these for standardization based on frequency of occurrence.
- o Patterns identify personnel skill sets and best practice training needed for more rapid/effective implementations. Identify both application and infrastructure patterns training so that individuals have a holistic perspective on the solution.
- o Prioritize and target improvements for quicker impact to business.

*Enhancement to Curriculum:* As an enterprise-wide pattern mining effort, this had significant impact to curriculum as it helped identify content requirements for a new professional master's tract in IT Services.  We list the highlights below:

- Enterprise Architecture presentations/communications training materials.
- Enterprise Architecture template and standard method to capture functional and non-functional materials
- Mined Patterns and Pattern taxonomy - Fourteen infrastructure and application patterns mined and organized from six projects.  These patterns are organized and illustrated (along with other related ones) in the diagram above.
- Pattern Socialization Objectives and Roles and Responsibilities for maintaining and adapting the standard architecture.
- Examples of Information Management Project Governance.

**Framework: "Vertical" or internal service-chain of primary business functions and enabling shared services**




**Framework results - Mined patterns that relate to the dimensions of the typical business domain on the left.**

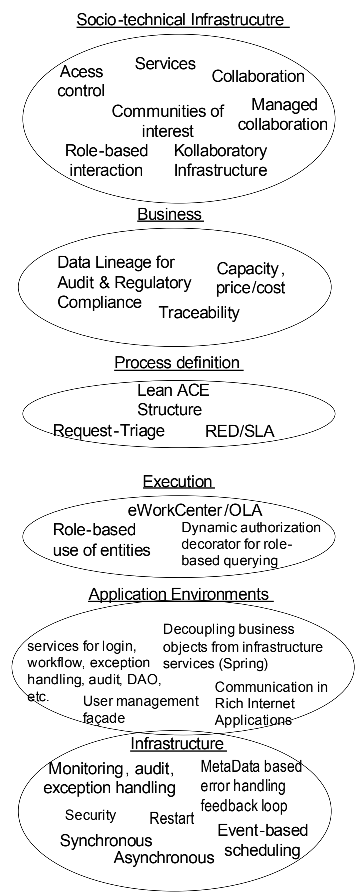



**Figure 8:  The internal dimensions and value-chain relationships between business and IT services, and related patterns.**

**Project:** Capacity Planning and Self-Tuning Techniques in Enterprise IT Data Centers.

**Problem and analysis:**  Our experiences with several consulting projects with the local industry in Mumbai have revealed a lack of knowledge of using performance analysis principles for system configuration in order that the appropriate service levels are met.

**Learning outcome:** Students should be provided with performance analysis and capacity management skills that they can apply directly upon graduation.

**Consequences:**

*Mined Knowledge Assets*: We have developed a taxonomy of performance analysis techniques of enterprise systems followed by tuning/configuration principles for the same.

*Impact to organization:* We have applied the techniques codified out of our experiences in data center management at several enterprise data centers and initiated a performance analysis lab here at IIT Bombay that has the tools needed for students to learn and experiment with performance analysis and capacity planning techniques

*Enhancements to Curriculum*: We have evolved a graduate course on self-tuning around the knowledge framework that we created. This course has contributions from the industry research labs as well who have been involved in creating some of this knowledge.

**Project:** Global Capstone Projects

**Problem and analysis:** As IT outsourcing becomes the norm, new ways and means of dealing with global multi-site development need to be developed. We have been conducting a set of capstone SE courses at OSU and IIT Bombay that present opportunities for students in these courses to work together to simulate outsourcing and global development scenarios. We have used the knowledge gained out of issues faced during this simulation exercise to enhance the SE processes for global development.

**Learning Outcome:** A codification of enhancements to software engineering processes for global development.

**Consequences:**

*Mined Knowledge Assets:* Assets consisted of guides [IBM97] on software engineering methodologies.

*Knowledge Framework:* A classification of global software engineering issues was presented in [Bellur06, Ramnath06] presented at the COMPSAC 2006 panel on Global Software Engineering.

*Enhancement to Curriculum:* Enhancements to the Software Engineering course at IIT Bombay.

## 5. Concluding Remarks

In this paper we have taken an approach for curriculum development that leverages "task conscious" methods for acquisition to develop "learning conscious" or formalized knowledge for teaching in the classroom. Given the applied nature of SE, we approach the curriculum development from the point of view that a) SE is an area of professional practice, b) the capacity to effectively reflect on action in a process of continuous learning is one of the defining characteristics of capable Professors of Practice that develop new practice knowledge, c) that key to effective reflection is the use of *knowledge frameworks* to define and frame the complex problems encountered, as well as to define solutions to these problems, and thus, d) our goal as educators must be to identify, validate, enhance, develop the necessary frameworks through communities of practice and then incorporate them into our curriculum.

### 5.1. Benefits

Our experience documented through examples herein indicates that the reflective pedagogy has several benefits:

- Starting with the process of identifying existing assets implies that any best-practice development indeed produces curricular results of long-term value.
- The time taken to identify missing assets and incrementally address them to deploy solutions has in many cases shown to have both a short response time (one to two years) and high impact for industry. At the same time it has yielded research formulation and results that would not have otherwise been feasible.
- The value of existing and new assets is continuously evaluated and improved.
- The process of reflective practice has served as process of skill renewal for teaching-research-focused faculty, something that has not been "institutionalized" yet within the framework of academia.

### 5.2. Contributions to Enterprise Transformation and Innovation

An understanding of larger trends within which IT departments have to function, have helped focus and target curriculum knowledge mining, how to develop knowledge, how to establish relevance and achieve success within shorter time frames than is currently the case. In summarizing a cross section of examples we note that each practice project lead to the discovery of new assets and/or a better integration of assets into the curriculum. We feel the success can be attributed in large part to the fact that the projects are selected as 'improvement' projects at the boundary of the '*known and the unknown*'.

We have also made a two-part business case to industry in order to gain their acceptance of this approach.

- The first is that their complex problems are being addressed using the set of new and potentially innovative best practices, and because of the pedagogical focus of each project, their personnel understand these new approaches in a deep manner, thus increasing their organization's overall capability.
- The second is that we have been able to show them that the students armed with these frameworks are better able to understand real industry problems better, and that industry is directly contributing to a better-trained workforce. An implicit benefit that our sponsors appear to appreciate is a closer connection with highly prominent Universities and a concrete way of influencing our curriculum, in a way that preserves academic integrity.

Even though this approach has been effective, it has taken time for results to emerge. Results have begun to emerge from Capstone projects in 2-year cycles, and in industry projects in 1-year cycles. Getting sponsorship of and willingness to participate in this approach from industry has been difficult, but we are slowly winning over partners.

In general we have been able to shown that, in addition to curriculum development, applied research and technology transfer results have also emerged. We believe that this model is viable and scalable, i.e. suitable for wider adoption by academic institutions by establishing the role of Professors of Practice.

### 5.3. Conclusions

In short, "business-as-usual" approaches are *not* a suitable model for providing curriculum targeted at the needs of professional *practice areas such as SE*. Academia in computing has become increasingly perceived as having less direct benefit; thus computing programs are down graded in value, and are suffering from declining and less diverse enrollments.

Capstone courses, internships and co-ops that give students industry experiences are nothing new. The uniqueness of our approach is that we have specifically identified industry projects for execution by undergraduate, graduate and faculty levels and targeted at developing and refining both University curriculum and Industry best practices within the Technology Using community. Also fairly unique is the scale of this initiative (6 projects a class, and almost 10 projects a year with industry per faculty). Another useful contribution is a perspective on what is needed in terms of future research work – IT-enabled services and the impact of globalization in the software industry and its impact on the SE processes.

The process described is an effective means for validating, enhancing and developing curriculum. Thus the curriculum assets created will certainly prove valuable to Universities engaged in developing effective SE programs, as well as to students and professionals in the field. Also, the establishment and "institutionalization" of reflective practice will serve as process of skill renewal for teaching-research-focused faculty. Finally, this will build an effective long-term partnership with industry that will have impact from the hiring and retention of graduates to research partnerships with faculty.

## 6. Acknowledgements

Our thanks go to the industry partners that collaborated with us on the projects mentioned in this paper. These include Gary Cavin (CIO), Jeffrey Clouse and Paul Carlson, as well as all the staff at the Department of Technology at the City of Columbus, Gouri Das at McGraw-Hill Education, Peter Shelkin (CTO) and Michael Ross at the Ohio State University Medical Center, David Pike, Madan Mohan and John Northland at Nationwide Insurance, and industry sponsors of our Capstone projects. Our thanks also go to Prof. Anish Arora, Dept. of Computer Science and Engineering at The Ohio State University and Principal Investigator of the ExScal project and DARPA. Finally, we thank Krista Dombroviak who contributed the taxonomy for pervasive computing applications, and members of the CETI Colloquium for their valuable feedback on this paper.

## 7. References

[Arora05] Arora, A., Ramnath, R., Ertin, E., Sinha, P. and others (26) "ExScal: Elements of an Extreme Scale Wireless Sensor Network," 11$^{th}$ IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, Hong Kong, China, 2005.

[Bass03] Bass, L., Clements, P., Kazman, R. Software Architecture in Practice, Addison-Wesley, 2003.

[Beck04] Beck, K., Andres, C., Extreme Programming Explained: Embrace Change (2nd Edition) (The XP Series), Addison-Wesley 2004.

[Bellur06] Bellur, U., An Academic Perspective on Globalization in the Software Industry, COMPSAC 2006, Chicago, USA, Sept 2006

[Boehm03] Boehm, B., Turner, R. Balancing Agility and Discipline, A Guide to the Perplexed, Addison-Wesley Longman Publishing Co., Inc., Boston, 2003.

[CMG] The Capacity Management Group http://www.cmg.org/

[Cassidy98] A Practical Guide to Information Systems Strategic Planning, Second Edition, CRC Press, 1998.

[CC2005] Computing Curricula 2005 http://www.computer.org/portal/pages/ieeecs/education/cc2001/index.html.

[George02] M. L. George, “Lean Six Sigma,” McGraw Hill, ISBN O-07-138521 (2002).

[Chandrasekharan99] Chandrasekharan, B., Josephson, J., Benjamins, V.R., What Are Ontologies and Why Do We Need Them, IEEE Intelligent Systems, 1999.

[Denning 2003] Denning, P. J., Great Principles of Computing, CACM Nov 2003.

[Chung00] Chung, L., Nixon, B. A., Yu, E. and Mylopoulos, J. Non-Functional Requirements in Software Engineering, Kluwer Academic Publishers, Boston, 2000.

[Exscal04] The Extreme-Scale Sensor Networking Project - http://www.cse.ohio-state.edu/exscal

[Fowler02] Fowler, M., Rice, D., Foemmel, M., Hieatt, E., Mee, R., Stafford, R., Patterns of Enterprise Application Architecture, Addison Wesley Professional, 2002

[Gamma95] Gamma, E., Helm, R., Johnson, R., Vlissides, J., Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley Professional Computing Series, 1995.

[Hammer93] M. Hammer and J. Champy, “Reengineering the Corporation: A Manifesto for Business Revolution,” Harper Business (1993).

[Haeckel99] Stephan H. Haeckel, Adaptive Enterprise: Creating and Leading Sense-And-Respond Organizations, Harvard Business School Press; (July 1, 1999), # ISBN: 0875848745.

[Highsmith04] Highsmith, J., Agile Project Management: Creating Innovative Products, Addison Wesley Professional, 2004

[ITIL] The IT Infrastructure Library http://www.itil.co.uk/

[IBM97] IBM Object-Oriented Technology Center, Developing Object-Oriented Software: An Experience-Based Approach, Prentice-Hall Publications, IBM Press, 1997.

[IBMSSME] IBM Services Science, Management and Engineering http://www.research.ibm.com/ssme/

[Jaccheri06] Jaccheri, L., Morasca, S. On the Importance of Dialogue with Industry about Software Engineering Education, Summit on Software Engineering Education, Shanghai, China, May 2006.

[Jones 03] D. T. Jones, J. P. Womack, "Lean Thinking: Banish Waste and Create Wealth in Your Corporation," Revised and Updated, Free Press; 2nd edition, ISBN: 0743249275 (2003).

[Kaplan96] Kaplan, R. S., Norton, D. P. The Balanced Scorecard: Translating Strategy into Action, Harvard Business School Press, Boston Mass., (1996).

[Mead00] Mead, N.R.; Saiedian, H.; Ruhe, G.; Bagert, D.J., Panel: shortages of qualified software engineering faculty and practitioners: challenges in breaking the cycle,. Proceedings of the International Conference on Software Engineering, ICSE 2000.

[Mockus01] Mockus, A., Weiss, D. M. Globalization by chunking: A quantitative approach. IEEE Software, 18(2), March 2001.

[Poppendieck03] Poppendieck, M., Poppendieck, T., Lean Software Development: An Agile Toolkit, Addison Wesley Professional, 2003.

[Porter79] Porter, M. E., How Competitive Forces Shape Strategy, Harvard Business Review, March-April 1979.

[Ramnath05] Ramnath, R., Landsbergen, D., IT-enabled sense-and-respond strategies in complex public organizations, Communications of the ACM, Volume 48, Issue 5 (May 2005), SPECIAL ISSUE: Adaptive Complex Enterprises.

[Ramanathan05] Ramanathan, J. "Fractal Architecture for the Adaptive Complex Enterprise", Communication of the ACM, Volume 48, Issue 5 (May 2005), SPECIAL ISSUE: Adaptive Complex Enterprises.

[Ramnath06] Ramnath, R, Global Software Development for the Enterprise, Invited Position Paper, Panel on Global Software Engineering Projects, IEEE Computer Society Signature Conference on Software Technology and Applications (COMPSAC), Chicago, Illinois, 2006.

[Ramnath05TR] Ramnath, R., Carlson, P., Desai, A., Using the NFR Framework in the context of the Balanced Scorecard: The Strategic Engineering of Public IT, OSU-CISRC-11/06-TR78, 2005.

[Ramanathan 07] Ramanathan, J., Glassgow, R., Ramnath, R. An Adaptive Complex Enterprise Framework for Lean Information Technology Enabled Services Delivery. OSU-CISRC-5/07-TR37. The Ohio State University. 2007.

[Rodriguez04] Managing software requirements: organizational and political challenges Authors: Rodriguez, Laurie L., 2004- Advisor: Charles P. Coleman. Department: Massachusetts Institute of Technology. Dept. of Aeronautics and Astronautics. Massachusetts Institute of Technology. Technology and Policy Program. http://hdl.handle.net/1721.1/17784.

[Schon83] Schon, D. A, "The Reflective Practitioner," New York: Basic Books, 1983

[Schon87] Schon, D.A, "Educating the Reflective Practitioner," Jossey-Bass Higher Education Series, John Wiley and Sons, 1987

[SEIPLA] Software Engineering Institute Product Lines Homepage http://www.sei.cmu.edu/productlines/

[STANDISH] The Chaos Report, Standish Group, www.standishgroup.com.

[SWEBOK04] The Software Engineering Body of Knowledge, http://www.swebok.org, 2004.

[TOGAF] http://www.togaf.org/